# UNITED STATES PATENT AND TRADEMARK OFFICE

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/647,058 | 08/21/2003 | J. Patrick Thompson | MSFT-1748/302722.01 | 1588 |

41505          7590          11/15/2007
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

| EXAMINER |
|---|
| PHAM, MICHAEL |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2167 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 11/15/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| | Application No. | Applicant(s) |
|---|---|---|
| **Office Action Summary** | 10/647,058 | THOMPSON ET AL. |
| | Examiner | Art Unit |
| | Michael D. Pham | 2167 |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1) ☒ Responsive to communication(s) filed on <u>04 September 2007</u>.

2a) ☒ This action is **FINAL.** 2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4) ☒ Claim(s) <u>1-11,13-21,23-25,27-31 and 33-60</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) <u>1-11,13-21,23-25,27-31 and 33-60</u> is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All b) ☐ Some * c) ☐ None of:

        1. ☐ Certified copies of the priority documents have been received.

        2. ☐ Certified copies of the priority documents have been received in Application No. _____.

        3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date <u>7/9/07</u>.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

U.S. Patent and Trademark Office

PTOL-326 (Rev. 08-06)          Office Action Summary          Part of Paper No./Mail Date 20071109

## DETAILED ACTION

### *Status of claims*

1.      Claims 1, 21, 37, 43, 49, and 55 have been amended.

2.      Claims 1-11, 13-21, 23-25, 27-31, and 33-60 are pending.

### *Specification*

3.      Prior objections are respectfully withdrawn.

### *Claim Rejections - 35 USC § 112*

4.      The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5.      Claims 1—20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Applicant's claim states that it is "...operable to belong..." in claim 1 line 5. It is unclear what Applicant's intended metes and bounds of the claim are, since the claim appears to cover anything and everything that does not prohibit actions from occurring. All other claims are rejected for failing to resolve the deficiencies of claim 1.

6.      Claims 21, 23-25, 27-31, and 33-36 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Applicant's claim states that it is "...capable of.." and "...operable to belong..." in claim 21 line 5. It is unclear what Applicant's intended metes and

bounds of the claim are, since the claim appears to cover anything and everything that does not

prohibit actions from occurring.   All other claims are rejected for failing to resolve the

deficiencies of claim 21.

### *Claim Rejections - 35 USC § 103*

6.      The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or
> described as set forth in section 102 of this title, if the differences between the subject
> matter sought to be patented and the prior art are such that the subject matter as a whole
> would have been obvious at the time the invention was made to a person having ordinary
> skill in the art to which said subject matter pertains.  Patentability shall not be negatived
> by the manner in which the invention was made.

7.      **Claims 1, 2, 3, 5, 7, 8, 10, 21, 37, 38, 39, 41, 43, 44, 45, 47, 49, 50, 51, 53, 55, 56, 57,**

**and 59 are rejected under 35 U.S.C. 103(a) as being unpatentable over US patent**

**application 2004/0199521 by Anglin et. al. (hereafter Anglin) further in view of US Patent**

**7158962 by Nelson (hereafter Nelson).**

### Claim 1:

Regarding claim 1, Anglin discloses the following claimed limitations:

" a plurality of Items" [storage object] ", where each of said plurality of Items constitute a

discrete storable unit of information that can be manipulated by a hardware/software interface

system" [paragraph [0020] FIG. 2a illustrates the data structure of a storage object entry or

record in the storage database that is added whenever a storage object is confirmed as written to

the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database. In Anglin, the "Item" of the claim is referred to as a 'storage object'];

"a plurality of Item Folders that constitute an organizational structure for said Items," [paragraph [0024] Storage objects [items] may be defined as part of a storage group [item folder] by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. In Anglin, the "Item Folder" of the claim is referred to as a "storage group"]

", each of said Items operable to belong to at least one of the Item Folders" [paragraph [0024] Moreover, with the described implementations, each storage object is capable of being associated with a plurality of different group types.]; and


"a hardware/software interface system for manipulating said plurality of Items." [paragraph [0018], The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art.].


Anglin does not explicitly disclose, "wherein said Item Folders are themselves Items and said Items are configured to be manipulated from any Item Folder they belong to" and "a plurality of Categories that constitute an additional organizational structure for said Items, at least one of

said Items belonging to at least one of the Categories, the Item Folders and the Categories

arranged in a directed graph structure".

Nelson discloses the following claimed limitations,

"wherein said Item Folders are themselves Items and said items are configured to be

manipulated from any Item Folder they belong to;"[ col. 1 lines 28-31, a folder is a named

collection of related items that can be retrieved, moved, and otherwise manipulated as one entity.

Col. 3 lines 59-60, an item can be for example a folder or document. Accordingly, each of said

Items (col. 1 lines 28-31, items) operable to belong to a plurality of Items Folders (col. 1 lines

28-31, collection of related items), wherein said Item Folders are themselves Items (col. 3 lines

59-60, an folders are also Items) and said items are configured to be manipulated from any Item

Folder they belong to (col. 1 lines 28-31, collection of related items that can be retrieved, moved,

and otherwise manipulated as one entity) is suggested.]

"a plurality of Categories that constitute an additional organizational structure for said

Items, at least one of said Items belonging to at least one of the Categories, the Item Folders and

the Categories arranged in a directed graph structure; and"[col. 1 lines 30-33, foldering or linking

is a process where a content management system manages and/or controls the creation, retrieval,

editing, and distribution of content within an information system. Foldering or linking is the act

of forming a link relationship between two Items. Col. 1 lines 38-42, This relationship is

maintained in a table known as a links table. Accordingly, a plurality of categories (col. 1 lines

38-39, each item in the system may exist as the source and/or target for linking) that constitute an

additional organizational structure for said Items (col. 1 lines 33-34, foldering or linking is the

act of forming a link relationship), at least one of said Items belonging to at least one of the Categories (col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking), the Item Folders and the Categories arranged in a directed graph structure (figure 3A) is suggested.]

Anglin and Nelson are all related to managing objects and are therefore within the same field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of Anglin in order to improve it by supporting multiple levels of folders and multiple attributes. Hence automating the tasks of linking items to folders and improving performance of multiple level nested folders.

## Claim 2:

Anglin discloses "wherein an Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

## Claim 3:

Anglin discloses "wherein an Item is automatically deleted when it no longer belongs to any Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

## Claim 5:

Anglin discloses "wherein said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

## Claim 7:

Anglin discloses "wherein each Item is a member of at least one Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of an Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group

ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

## Claim 8:

Anglin discloses "wherein each said Item is itself automatically deleted when it no longer belongs to any Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

## Claim 10:

Anglin discloses "wherein each said Item is itself automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

## Claim 21:

Anglin discloses the following claimed limitations:

"a hardware/software interface system capable of manipulating an Item of a plurality of

Items"(data objects) [paragraph 0018, the Server includes a storage management server program

that is capable of performing storage related operations of data objects received from data

management client programs. The storage management operations may comprise backup

operations, archival operations, hierarchical storage management related operations or any type

of storage management operations known in the art. Because 'data objects' is plural, it is

interpreted that there are a plurality of items.];

"said Item comprising a discrete unit of information comprising a basic set of properties

commonly supported across objects exposed by an operating system shell" [paragraph 0020,

FIG. 2a illustrates the data structure of a storage object entry or record in the storage database

that is added whenever a storage object is confirmed as written to the storage. The entry includes

a unique identifier that uniquely identifies the storage object and entry in the storage database. In

Anglin et al., the "Item" of the claim is referred to as a "storage object"]

"said Item" (objects) "being a fundamental unit of information manipulated by an

operating system" [paragraph 0005, in the prior art, the storage management program may define

a group as specific objects to be managed, such as data files, database files, programs, etc. This

list of objects define fundamental units of information.]

", said Item operable to belong to a plurality of Item Folders" (groups) [paragraph 0024,

moreover, with the described implementations, each storage object is capable of being associated

with a plurality of different group types. And paragraph 0019, if the storage objects are

associated with an object group, then indication of the associated object group would be included

with the object information in the storage database. In the reference, the "Item" of the instant

application is called the "storage object" and the "Item folder" of the instant application is called

the "object group" or in some cases the "storage group"].

Anglin does not explicitly disclose, "wherein said Item Folders are themselves Items and

said Items are configured to be manipulated from any Item Folder they belong to, and said Item

configured to be a member of one of a plurality of Categories that constitute an organizational

structure for said Items, the Items Folders and the Categories arranged in a directed graph

structure"

Nelson discloses the following claimed limitations,

"wherein said Item Folders are themselves Items and said Items are configured to be

manipulated from any Item folder they belong to," [ col. 1 lines 28-31, a folder is a named

collection of related items that can be retrieved, moved, and otherwise manipulated as one entity.

Col. 3 lines 59-60, an item can be for example a folder or document. Accordingly, wherein said

Item Folders are themselves Items (col. 3 lines 59-60, an folders are also Items) and said items

are configured to be manipulated from any Item Folder they belong to (col. 1 lines 28-31,

collection of related items that can be retrieved, moved, and otherwise manipulated as one entity)

is suggested.]

"and said Item configured to be a member of one of a plurality of Categories that

constitute an organizational structure for said Items," [col. 1 lines 30-33, foldering or linking is a

process where a content management system manages and/or controls the creation, retrieval,

editing, and distribution of content within an information system. Foldering or linking is the act

of forming a link relationship between two Items. Col. 1 lines 38-42, This relationship is

maintained in a table known as a links table. Accordingly, said Item configured to be a member

of one of a plurality of Categories (col. 1 lines 38-39, each item in the system may exist as the

source and/or target for linking. This relationship is maintained in a table known as a Links

Table) that constitute an organizational structure for said Items (Foldering or linking is the act of

forming a link relationship between two Items) is suggested.]

"the Item Folders and the Categories arranged in a directed graph structure." [col. 1 lines

30-33, foldering or linking is a process where a content management system manages and/or

controls the creation, retrieval, editing, and distribution of content within an information system.

Foldering or linking is the act of forming a link relationship between two Items. Col. 1 lines 38-

42, This relationship is maintained in a table known as a links table. Accordingly, a plurality of

categories (col. 1 lines 38-39, each item in the system may exist as the source and/or target for

linking) that constitute an additional organizational structure for said Items (col. 1 lines 33-34,

foldering or linking is the act of forming a link relationship), at least one of said Items belonging

to at least one of the Categories (col. 1 lines 38-39, each item in the system may exist as the

source and/or target for linking), the Item Folders and the Categories arranged in a directed graph

structure (figure 3A) is suggested.]


Anglin and Nelson are all related to managing objects and are therefore within the same

field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in

the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of

Anglin in order to improve it by supporting multiple levels of folders and multiple attributes.

Hence automating the tasks of linking items to folders and improving performance of multiple

level nested folders.

**Claim 37:**

Anglin discloses the following claimed limitations

"a method for organizing Items in computer system, said Items comprising a discrete unit

of information that can be manipulated by a hardware/software interface system" [paragraph

[0018] The Server includes a storage management server program that is capable of performing

storage related operations of data objects received from data management client programs. The

storage management operations may comprise backup operations, archival operations,

hierarchical storage management related operations or any type of storage management

operations known in the art];

"said method comprising means by which an Item can be a member of at least two Item

Folders" [paragraph [0026] Many type of storage object management operations would have to

take into account group characteristics and that a storage object may be a member of multiple

group types.];

"but is not owned by any of said Item Folders such that the deletion of any of said Item

Folders does not automatically result in the deletion of said Item" [paragraph [0028] The

secondary deletion ensure that a storage object and corresponding storage object entry are only

removed if the storage object is not a member of any further groups after eliminating the

relationship between the storage object and group i. This follows the request to delete the "group

leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.]

Anglin does not explicitly disclose "means by which said Item can be a member of a Category of a plurality of Categories that constitute an organizational structure for said Items, the Item Folders and the Categories arranged in a directed graph structure, wherein said Item folders are themselves Items and said items are configured to be manipulated from any Item folder they belong to."

Nelson discloses "and means by which said item can be a member of a category of a plurality of categories that constitute an organizational structure for said items," [col. 1 lines 30-33, foldering or linking is a process where a content management system manages and/or controls the creation, retrieval, editing, and distribution of content within an information system. Foldering or linking is the act of forming a link relationship between two Items. Col. 1 lines 38-42, This relationship is maintained in a table known as a links table. Accordingly, means by which said item can be a member of a category of a plurality of categories (col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking) that constitute an additional organizational structure for said Items (col. 1 lines 33-34, foldering or linking is the act of forming a link relationship), the item folders and categories arranged in a directed graph structure (figure 3A) is suggested.]

"wherein said item folders are themselves items and said items are configured to be manipulated from any item folder they belong to." [ col. 1 lines 28-31, a folder is a named

collection of related items that can be retrieved, moved, and otherwise manipulated as one entity. Col. 3 lines 59-60, an item can be for example a folder or document. Accordingly, wherein said Item Folders are themselves Items (col. 3 lines 59-60, an folders are also Items) and said items are configured to be manipulated from any Item Folder they belong to (col. 1 lines 28-31, collection of related items that can be retrieved, moved, and otherwise manipulated as one entity) is suggested.]

Anglin and Nelson are all related to managing objects and are therefore within the same field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of Anglin in order to improve it by supporting multiple levels of folders and multiple attributes. Hence automating the tasks of linking items to folders and improving performance of multiple level nested folders.

**Claim 38:**

Anglin discloses "wherein the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

**Claim 39:**

Anglin discloses "wherein the Item is automatically deleted when it no longer belongs to any

Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in

the associated groups fields for the specified storage object entry, then the storage management

server deletes the specified storage object entry from the storage database and deletes the

identifier of the deleted storage object entry from the group entry for the target group.")


**Claim 41:**

Anglin discloses "The method of claim 38 wherein the Item is automatically deleted when it is a

member of only one Item Folder and said Item Folder is deleted." (See page 3, paragraph [0028]

"The secondary deletion ensure that a storage object and corresponding storage object entry are

only removed if the storage object is not a member of any further groups after eliminating the

relationship between the storage object and group i." This follows the request to delete the

"group leader" which represents the group ID of the "storage group" or the "Item Folder" as in

referred to in the claim.)


**Claim 43:**

Anglin discloses the following claimed limitations:

       "a computer-readable medium comprising computer-readable instructions for an Item of

a plurality of Items, said Items comprising a discrete unit of information that can be manipulated

by a hardware/software interface system" (See page 2, paragraph [0018] "The Server includes a

storage management server program that is capable of performing storage related operations of

data objects received from data management client programs. The storage management

operations may comprise backup operations, archival operations, hierarchical storage

management related operations or any type of storage management operations known in the

art...")

"a plurality of Item Folders that constitute an organizational structure for said Items,"

(See page 2, paragraph [0024] "Storage objects [items] may be defined as part of a storage group

[item folder] by adding the group identified of the one or more groups in which the storage

object is a member to the group field of the storage object entry." In Anglin, the "Item Folder"

of the claim is referred to as a "storage group"),

"each of said Items belonging to at least one of the Item Folders" (See page 2, paragraph

[0024] "Moreover, with the described implementations, each storage object is capable of being

associated with a plurality of different group types.")

Anglin however does not explicitly disclose, "wherein said Item Folders are themselves

Items and said items are configured to be manipulated from any Item Folder they belong to; said

Item being a member of one of a plurality of Categories that constitute an organizational

structure for said Items, the Categories arranged in a directed graph structure."

"wherein said item folders are themselves items and said items configured to be

manipulated from any item folder they belong to;" [ col. 1 lines 28-31, a folder is a named

collection of related items that can be retrieved, moved, and otherwise manipulated as one entity.

Col. 3 lines 59-60, an item can be for example a folder or document. Accordingly, wherein said

Item Folders are themselves Items (col. 3 lines 59-60, an folders are also Items) and said items

are configured to be manipulated from any Item Folder they belong to (col. 1 lines 28-31,

collection of related items that can be retrieved, moved, and otherwise manipulated as one entity)

is suggested.]

"and said item being a member of one of a plurality of Categories that constitute an

organizational structure for said Items, the categories arranged in a directed graph structure" [col.

1 lines 30-33, foldering or linking is a process where a content management system manages

and/or controls the creation, retrieval, editing, and distribution of content within an information

system. Foldering or linking is the act of forming a link relationship between two Items. Col. 1

lines 38-42, This relationship is maintained in a table known as a links table. Accordingly, said

item being a member of one of a plurality of categories (col. 1 lines 38-39, each item in the

system may exist as the source and/or target for linking) that constitute an additional

organizational structure for said Items (col. 1 lines 33-34, foldering or linking is the act of

forming a link relationship), at least one of said Items belonging to at least one of the Categories

(col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking), the

Categories arranged in a directed graph structure (figure 3A) is suggested.]

Anglin and Nelson are all related to managing objects and are therefore within the same

field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in

the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of

Anglin in order to improve it by supporting multiple levels of folders and multiple attributes.

Hence automating the tasks of linking items to folders and improving performance of multiple

level nested folders.

## Claim 44:

Anglin discloses "wherein the Item is a member of an Item Folder but is not owned by said Item

Folder, such that the deletion of said Item Folder does not automatically result in the deletion of

said Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object

and corresponding storage object entry are only removed if the storage object is not a member of

any further groups after eliminating the relationship between the storage object and group i."

This follows the request to delete the "group leader" which represents the group ID of the

"storage group" or the "Item Folder" as in referred to in the claim.)

## Claim 45:

Anglin discloses "The computer-readable medium of claim 44 wherein the Item is automatically

deleted when it no longer belongs to any Item Folder" (See page 3, paragraph [0029] "If the

target group is the only indicated group in the associated groups fields for the specified storage

object entry, then the storage management server deletes the specified storage object entry from

the storage database and deletes the identifier of the deleted storage object entry from the group

entry for the target group.").

## Claim 47:

Anglin discloses, "The computer-readable medium of claim 44 wherein the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

## Claim 49:

Anglin discloses the following claimed limitations:

"a computer readable medium comprising computer-readable instructions for organizing Items in computer system, said Items comprising a discrete unit of information that can be manipulated by a hardware/software interface system" [paragraph 0018, The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art];

"said method comprising means by which an Item can be a member of at least two Item Folders" [paragraph 0026, Many type of storage object management operations would have to take into account group characteristics and that a storage object may be a member of multiple group types.];

"but is not owned by any of said Item Folders such that the deletion of any of said Item

Folders does not automatically result in the deletion of said Item" [paragraph 0028, the

secondary deletion ensure that a storage object and corresponding storage object entry are only

removed if the storage object is not a member of any further groups after eliminating the

relationship between the storage object and group i. This follows the request to delete the "group

leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred

to in the claim.]

Anglin does not explicitly disclose "means by which said Item can be a member of a Category of

a plurality of Categories that constitute an organizational structure for said Items, the Items

Folders and the Categories arranged in a directed graph structure, wherein said Item Folders are

themselves Items and said Items are configured to be manipulated from Item Folder they belong

to."

On the other hand, nelson discloses:

"and means by which said item can be a member of a category of a plurality of categories

that constitute an organizational structure for said items, the item folder and the categories

arranged in a directional graph structure" [col. 1 lines 30-33, foldering or linking is a process

where a content management system manages and/or controls the creation, retrieval, editing, and

distribution of content within an information system. Foldering or linking is the act of forming a

link relationship between two Items. Col. 1 lines 38-42, This relationship is maintained in a

table known as a links table. Accordingly, means by which said item can be a member of a

category of a plurality of categories (col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking) that constitute an organizational structure for said Items (col. 1 lines 33-34, foldering or linking is the act of forming a link relationship), the item folders and categories arranged in a directed graph structure (figure 3A) is suggested.]

"wherein said item folders are themselves items and said items are configured to be manipulated from item folder they belong to." [ col. 1 lines 28-31, a folder is a named collection of related items that can be retrieved, moved, and otherwise manipulated as one entity. Col. 3 lines 59-60, an item can be for example a folder or document. Accordingly, wherein said Item Folders are themselves Items (col. 3 lines 59-60, an folders are also Items) and said items are configured to be manipulated from any Item Folder they belong to (col. 1 lines 28-31, collection of related items that can be retrieved, moved, and otherwise manipulated as one entity) is suggested.]

Anglin and Nelson are all related to managing objects and are therefore within the same field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of Anglin in order to improve it by supporting multiple levels of folders and multiple attributes. Hence automating the tasks of linking items to folders and improving performance of multiple level nested folders.

## Claim 50:

Anglin discloses "The computer-readable medium of claim 49 wherein the Item is a member of

an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder

does not automatically result in the deletion of said Item." (See page 3, paragraph [0028] "The

secondary deletion ensure that a storage object and corresponding storage object entry are only

removed if the storage object is not a member of any further groups after eliminating the

relationship between the storage object and group i." This follows the request to delete the

"group leader" which represents the group ID of the "storage group" or the "Item Folder" as in

referred to in the claim.)

## Claim 51:

Anglin discloses "The computer-readable medium of claim 50 wherein the Item is automatically

deleted when it no longer belongs to any Item Folder." (See page 3, paragraph [0029] "If the

target group is the only indicated group in the associated groups fields for the specified storage

object entry, then the storage management server deletes the specified storage object entry from

the storage database and deletes the identifier of the deleted storage object entry from the group

entry for the target group.")

## Claim 53:

Anglin discloses "The computer-readable medium of claim 50 wherein the Item is automatically

deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page

3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding

storage object entry are only removed if the storage object is not a member of any further groups

after eliminating the relationship between the storage object and group i." This follows the

request to delete the "group leader" which represents the group ID of the "storage group" or the

"Item Folder" as in referred to in the claim.)


**Claim 55:**

Anglin discloses the following claimed limitations:

"means for manipulating a plurality of Items comprising at least one Item, where each of

said plurality of Items constitute a discrete unit of information that can be manipulated by a

hardware/software interface system" [paragraph 0020, FIG. 2a illustrates the data structure of a

storage object entry or record in the storage database that is added whenever a storage object is

confirmed as written to the storage. The entry includes a unique identifier that uniquely

identifies the storage object and entry in the storage database. In Anglin, the "Item" of the claim

is referred to as a "storage object"];

"means for manipulating a plurality of Item Folders comprising at least one Item Folder,

wherein said plurality of Item Folders constitute an organizational structure for said Items"

[Paragraph 0024, Storage objects may be defined as part of a storage group by adding the group

identified of the one or more groups in which the storage object is a member to the group field of

the storage object entry. In Anglin, the "Item Folder" of the claim is referred to as a "storage

group"]; and

"wherein each of said plurality of Items belongs to at least one of said plurality of Item

Folders, and wherein each of said plurality of Items may belong to more than one Item Folder of

said plurality of Item Folders" [paragraph 0018, the Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art. And paragraph 0019, if the storage objects are associated with an object group, then indication of the associated object group would be included with the object information in the storage database. And paragraph 0024, by adding the group identifier of the one or more groups in which the storage object is a member – In other words, the "Item" ("storage object") may belong to more than one folder ("storage group")].

Anglin does not explicitly disclose "and said Item Folders are themselves Items and said Items are configured to be manipulated from any Item folder they belong to;" and "means for manipulating a plurality of Categories that constitute an additional organizational structure for said Items, at least one of said Items belonging to at least one of the Categories, the Item Folders and the Categories arranged in a directed graph structure".

Nelson discloses the following claimed limitations:

"and said item folders are themselves items and said items are configured to be manipulated from item folders they belong to; and" [col. 1 lines 28-31, a folder is a named collection of related items that can be retrieved, moved, and otherwise manipulated as one entity. Col. 3 lines 59-60, an item can be for example a folder or document. Accordingly, said Item Folders are themselves Items (col. 3 lines 59-60, an folders are also Items) and said items are

configured to be manipulated from any Item Folder they belong to (col. 1 lines 28-31, collection of related items that can be retrieved, moved, and otherwise manipulated as one entity) is suggested.]

"Means for manipulating a plurality of categories that constitute an additional organizational structure for said Items, at least one of said Items belonging to at least one of the categories, the item folders and the categories arranged in a directed graph structure," [col. 1 lines 30-33, foldering or linking is a process where a content management system manages and/or controls the creation, retrieval, editing, and distribution of content within an information system. Foldering or linking is the act of forming a link relationship between two Items. Col. 1 lines 38-42, This relationship is maintained in a table known as a links table. Accordingly, means for manipulating a plurality of categories (col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking) that constitute an additional organizational structure for said Items (col. 1 lines 33-34, foldering or linking is the act of forming a link relationship), at least one of said Items belonging to at least one of the categories (col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking), and the item folders and categories arranged in a directed graph structure (figure 3A) is suggested.]

Anglin and Nelson are all related to managing objects and are therefore within the same field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of Anglin in order to improve it by supporting multiple levels of folders and multiple attributes.

Hence automating the tasks of linking items to folders and improving performance of multiple level nested folders.

## Claim 56:

Anglin discloses "The computer-readable medium of claim 55 wherein an Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

## Claim 57:

Anglin discloses "The computer-readable medium of claim 56 wherein the Item is automatically deleted when it no longer belongs to any Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

## Claim 59:

Anglin discloses "The computer-readable medium of claim 56 wherein the Item is automatically

deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page

3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding

storage object entry are only removed if the storage object is not a member of any further groups

after eliminating the relationship between the storage object and group i." This follows the

request to delete the "group leader" which represents the group ID of the "storage group" or the

"Item Folder" as in referred to in the claim.)

**8.        Claims 4, 6, 9, 11, 40, 42, 46, 48, 52, 54, 58, and 60 rejected under 35 U.S.C. 103(a) as**

**being unpatentable over US patent application 2004/0199521 by Anglin et. al. (hereafter**

**Anglin) and US Patent 7158962 by Nelson (hereafter Nelson) further in view of US**

**20040073560 by Edwards (hereafter Edwards)**

## Claim 4:

Anglin, and Nelson teach a computer system substantially as claimed.  Anglin, and Nelson do

not explicitly teach an Item, when it no longer belongs to any Item Folder, automatically

becomes a member of a default Item Folder.  Nelson does discloses col. 5 lines 54-56, No folder

exists matching these attributes. Consequently, the library server 25 creates a folder with the

following attributes.   However, Edwards teaches more explicitly an Item, when it no longer

belongs to any Item Folder, automatically becomes a member of a default Item Folder.  (See

page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder). It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

## Claim 6:

Anglin and Nelson teach a computer system substantially as claimed. Anglin, and Nelson do not explicitly teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, Edwards teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be

'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin and Nelson. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

## Claim 9:

Regarding claim 9, Anglin and Nelson teach a computer system substantially as claimed. Anglin and Nelson do no explicitly teach each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder. Nelson does disclose col. 5 lines 54-56, No folder exists matching these attributes. Consequently, the library server 25 creates a folder with the following attributes. However, Edwards more explicitly teaches each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant

application, the area in which the item is stored is called the default Item Folder.) It would have

been obvious to one with ordinary skill in the art to combine the Item and Item folder system of

Anglin and Nelson with the disclosure of a recycling method of Edwards by simply adding the

recycling method of Edwards to the system of Anglin and Nelson. Edwards points out the

advantage of keeping the potentially deleted file (or in the instant application, the object,)

available for future use rather than immediate deletion. It is for this reason that one of ordinary

skill in the art would have been motivated to have each said Item, when each no longer belongs

to any Item Folder, automatically become members of a default Item Folder.


## Claim 11:

Regarding claim 11, Anglin and Nelson teach a computer system substantially as claimed.

Anglin and Nelson do not explicitly teach each said Item, when each is a member of only one

Item Folder and said Item Folder is deleted, automatically become members of a default Item

Folder. Nelson does disclose col. 5 lines 54-56, No folder exists matching these attributes.

Consequently, the library server 25 creates a folder with the following attributes. However,

Edwards teaches each said Item, when each is a member of only one Item Folder and said Item

Folder is deleted, automatically become members of a default Item Folder. (See page 3,

paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means

that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating

system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it

can be retrieved if required." In the instant application, the area in which the item is stored is

called the default Item Folder.) It would have been obvious to one with ordinary skill in the art

to combine the Item and Item folder system of Anglin and Nelson with the disclosure of a

recycling method of Edwards by simply adding the recycling method of Edwards to the system

of Anglin and Nelson. Edwards points out the advantage of keeping the potentially deleted file

(or in the instant application, the object,) available for future use rather than immediate deletion.

It is for this reason that one of ordinary skill in the art would have been motivated to have each

said Item, when each is a member of only one Item Folder and said Item Folder is deleted,

automatically become members of a default Item Folder.


### Claim 40:

Regarding claim 40, Anglin, and Nelson teach a method substantially as claimed. Anglin and

Nelson do not explicitly teach said Item, when it no longer belongs to any Item Folder,

automatically becomes a member of a default Item Folder. However, Edwards teaches said Item,

when it no longer belongs to any Item Folder, automatically becomes a member of a default Item

Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than

delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is

passed to the operating system to be 'recycled'. This means it is not deleted immediately, but

stored in an area where it can be retrieved if required." In the instant application, the area in

which the item is stored is called the default Item Folder). It would have been obvious to one

with ordinary skill in the art to combine the Item and Item folder system of Anglin and Nelson

with the disclosure of a recycling method of Edwards by simply adding the recycling method of

Edwards to the system of Anglin. Edwards points out the advantage of keeping the potentially

deleted file (or in the instant application, the object,) available for future use rather than

immediate deletion. It is for this reason that one of ordinary skill in the art would have been

motivated to have said Item, when it no longer belongs to any Item Folder, automatically

becomes a member of a default Item Folder.


## Claim 42:

Regarding claim 42, Anglin and Nelson teach a method substantially as claimed. Anglin and

Nelson do not explicitly teach said Item, when it is a member of only one Item Folder and said

Item Folder is deleted, automatically becomes a member of a default Item Folder. However,

Edwards teaches said Item, when it is a member of only one Item Folder and said Item Folder is

deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph

[0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that

whenever the synchroniser is to over-write or delete a file, the file is passed to the operating

system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it

can be retrieved if required." In the instant application, the area in which the item is stored is

called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to

combine the Item and Item folder system of Anglin and Nelson with the disclosure of a recycling

method of Edwards by simply adding the recycling method of Edwards to the system of Anglin

and Nelson. Edwards points out the advantage of keeping the potentially deleted file (or in the

instant application, the object,) available for future use rather than immediate deletion. It is for

this reason that one of ordinary skill in the art would have been motivated to have said Item,

when it is a member of only one Item Folder and said Item Folder is deleted, automatically

becomes a member of a default Item Folder.

### Claim 46:

Anglin and Nelson do not explicitly disclose "said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin and Nelson. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

## Claim 48:

Anglin and Nelson do not explicitly disclose "when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin and Nelson. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

## Claim 52:

Anglin and Nelson do not explicitly disclose "said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin and Nelson. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

## Claim 54:

Anglin and Nelson do not explicitly disclose "said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it is a member of only one Item Folder and

said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page

3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This

means that whenever the synchroniser is to over-write or delete a file, the file is passed to the

operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area

where it can be retrieved if required." In the instant application, the area in which the item is

stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in

the art to combine the Item and Item folder system of Anglin and Nelson with the disclosure of a

recycling method of Edwards by simply adding the recycling method of Edwards to the system

of Anglin and Nelson. Edwards points out the advantage of keeping the potentially deleted file

(or in the instant application, the object,) available for future use rather than immediate deletion.

It is for this reason that one of ordinary skill in the art would have been motivated to have said

Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically

becomes a member of a default Item Folder.


**Claim 58:**

Anglin and Nelson do not explicitly disclose "wherein said Item, when it no longer belongs to

any Item Folder, automatically becomes a member of a default Item Folder." However, Edwards

teaches an Item, when it no longer belongs to any Item Folder, automatically becomes a member

of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to

'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or

delete a file, the file is passed to the operating system to be 'recycled'. This means it is not

deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin and Nelson. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

**Claim 60:**

Anglin and Nelson do not explicitly disclose "said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin and Nelson with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system

of Anglin and Nelson. Edwards points out the advantage of keeping the potentially deleted file

(or in the instant application, the object,) available for future use rather than immediate deletion.

It is for this reason that one of ordinary skill in the art would have been motivated to have said

Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically

becomes a member of a default Item Folder.

9.      **Claims 13-20 and 33-36 are rejected under 35 U.S.C. 103(a) as being unpatentable**

**over US patent application 2004/0199521 by Anglin et. al. (hereafter Anglin) and US Patent**

**7158962 by Nelson (hereafter Nelson) further in view of US Patent 6578046 by Chang et. al.**

**(hereafter Chang)**

### Claim 13:

Anglin and Nelson do not explicitly disclose "The computer system of claim 1 wherein a

category is defined by an Item property." Nelson discloses col. 4 lines 60-62, automatic linking

rules for that item type. These rules are stored in a separate table in the library server. However,

Chang more explicitly teaches a Category is defined by an Item property. (See column 13, lines

26-30 "In the preferred embodiment, a FederatedCollection allows an application program to

process data objects resulting from a query as a group or collection and at the same time

preserves the sub-groupings relationships that exist between them." Simply by the Category

being formed as the result of a query, the Item property is necessarily what defined the Category,

as the data must meet the Item property in order to be a result of the query.) It would have been

obvious to one with ordinary skill the art to combine the system as disclosed in Anglin and

Nelson with the disclosure of Chang because it would be logical to use the Item property as what

defines the categories, especially in the case of a query.  It is for this reason that one of ordinary

skill in the art would have been motivated to have a Category is defined by an Item property.

**Claim 14:**

Nelson and Anglin do not explicitly disclose "wherein one of said plurality of Categories is

defined by an Item Property and only an Item comprising the Item property for a specific

Category from among said plurality of Categories can be a member of said specific Category."

The combination of Nelson and Anglin disclose in Nelson col. 4 lines 60-62, automatic linking

rules for that item type.  These rules are stored in a separate table in the library server.  However,

Chang teaches more explicitly wherein one of said plurality of Categories is defined by an Item

Property and only an Item comprising the Item property for a specific Category from among said

plurality of Categories can be a member of said specific Category. (See column 13, lines 26-30

"In the preferred embodiment, a FederatedCollection allows an application program to process

data objects resulting from a query as a group or collection and at the same time preserves the

sub-groupings relationships that exist between them."  Simply by the Category being formed as

the result of a query, the Item property is necessarily what defined the Category, as the data must

meet the Item property in order to be a result of the query.  Also, by the nature of queries only

returning the results that are related, the only members of the Category will be from results that

comprise the Item property.)  It would have been obvious to one with ordinary skill the art to

combine the system as disclosed in Anglin and Nelson with the disclosure of Chang because it

would be logical to use the Item property as what defines the categories, especially in the case of

a query. It is for this reason that one of ordinary skill in the art would have been motivated to

have one of said plurality of Categories is defined by an Item property, and only an Item

comprising the Item property for a specific Category from among said plurality of Categories can

be a member of said specific Category.

## Claim 15:

Regarding claim 15, Anglin and Nelson teaches a computer system substantially as shown.

Anglin and Nelson do not explicitly disclose an Item comprising the Item property for one of

said plurality of Categories is automatically a member of that one of said plurality of Categories.

However, Chang teaches an Item comprising the Item property for one of said plurality of

Categories is automatically a member of that one of said plurality of Categories. (See column 13,

lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program

to process data objects resulting from a query as a group or collection and at the same time

preserves the sub-groupings relationships that exist between them." By the nature of queries

only returning the results that are related, the only members of the Category will be from results

that comprise the Item property. Here, all of the results of the query are included in on the

collection [referred to in the instant application as a category].) It would have been obvious to

one with ordinary skill the art to combine the system as disclosed in Anglin and Nelson with the

disclosure of Chang because keeping the results of the query, all of which exhibit a relationship,

is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories.

## Claim 16:

Regarding claim 16, Nelson and Anglin teaches a computer system substantially as shown. Anglin and Nelson do not explicitly disclose an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties. However, Chang teaches an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin and Nelson with the disclosure of Chang because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run

again. It is for this reason that one of ordinary skill in the art would have been motivated to

include an Item comprising one or more Item properties corresponding to one or more Categories

of said plurality of Categories is automatically a member of each such Categories for said

corresponding Item properties.

**Claim 17:**

Anglin and Nelson do not explicitly discloses "wherein each of said plurality of categories is

defined by an item property." Nelson does disclose col. 4 lines 60-62, automatic linking rules for

that item type. These rules are stored in a separate table in the library server. However, Chang

more explicitly teaches each of said plurality of Categories is defined by an Item property. (See

column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an

application program to process data objects resulting from a query as a group or collection and at

the same time preserves the sub-groupings relationships that exist between them." Simply by the

Category being formed as the result of a query, the Item property is necessarily what defined the

Category, as the data must meet the Item property in order to be a result of the query.) It would

have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin

and Nelson with the disclosure of Chang because it would be logical to use the Item property as

what defines the categories, especially in the case of a query. It is for this reason that one of

ordinary skill in the art would have been motivated to have each of said plurality of Categories is

defined by an Item property.

## Claim 18:

Regarding claim 18, Anglin and Nelson teaches a computer system substantially as shown. Anglin and Nelson do not explicitly teach each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category. However, Chang teaches each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin and Nelson with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category.

## Claim 19:

Regarding claim 19, Anglin and Nelson teaches a computer system substantially as shown.

Anglin and Nelson do not explicitly teach each Item comprising the Item property for one of said

plurality of Categories are automatically members of that one of said plurality of Categories.

However, Chang teaches each Item comprising the Item property for one of said plurality of

Categories are automatically members of that one of said plurality of Categories. (See column

13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application

program to process data objects resulting from a query as a group or collection and at the same

time preserves the sub-groupings relationships that exist between them." By the nature of

queries only returning the results that are related, the only members of the Category will be from

results that comprise the Item property. Here, all of the results of the query are included in on

the collection [referred to in the instant application as a category].) It would have been obvious

to one with ordinary skill the art to combine the system as disclosed in Anglin and Nelson with

the disclosure of Chang because keeping the results of the query, all of which exhibit a

relationship, is useful in that the query will not have to be run again. It is for this reason that one

of ordinary skill in the art would have been motivated to include each Item comprising the Item

property for one of said plurality of Categories are automatically members of that one of said

plurality of Categories.

## Claim 20:

Regarding claim 20, Anglin and Nelson teaches a computer system substantially as shown.

Anglin and Nelson do not explicitly teach all Items comprising one or more Item properties

corresponding to one or more Categories of said plurality of Categories are automatically

members of all such Categories for said corresponding Item properties.    However, Chang

teaches all Items comprising one or more Item properties corresponding to one or more

Categories of said plurality of Categories are automatically members of all such Categories for

said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a

FederatedCollection allows an application program to process data objects resulting from a query

as a group or collection and at the same time preserves the sub-groupings relationships that exist

between them." By the nature of queries only returning the results that are related, the only

members of the Category will be from results that comprise the Item property.  Here, all of the

results of the query are included in on the collection [referred to in the instant application as a

category].)    It would have been obvious to one with ordinary skill the art to combine the system

as disclosed in Anglin and Nelson with the disclosure of Chang because keeping the results of

the query, all of which exhibit a relationship, is useful in that the query will not have to be run

again.  It is for this reason that one of ordinary skill in the art would have been motivated to

include all Items comprising one or more Item properties corresponding to one or more

Categories of said plurality of Categories are automatically members of all such Categories for

said corresponding Item properties.


**Claim 33:**

Anglin and Nelson do not explicitly discloses "The hardware/software interface system of claim

21 wherein said category is defined by an item property".  Nelson discloses col. 4 lines 60-62,

automatic linking rules for that item type. These rules are stored in a separate table in the library

server. However, Chang more explicitly teaches said Category is defined by an Item property.

(See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an

application program to process data objects resulting from a query as a group or collection and at

the same time preserves the sub-groupings relationships that exist between them." Simply by the

Category being formed as the result of a query, the Item property is necessarily what defined the

Category, as the data must meet the Item property in order to be a result of the query.) It would

have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin

and Nelson with the disclosure of Chang because it would be logical to use the Item property as

what defines the categories, especially in the case of a query. It is for this reason that one of

ordinary skill in the art would have been motivated to have said Category is defined by an Item

property.

## Claim 34:

Regarding claim 34, Anglin and Nelson teaches a hardware/software interface system

substantially as shown. Anglin and Nelson do not explicitly disclose said Category is defined by

an Item property, and only an Item comprising the Item property for said Category can be a

member of said Category. However, Chang teaches said Category is defined by an Item

property, and only an Item comprising the Item property for said Category can be a member of

said Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection

allows an application program to process data objects resulting from a query as a group or

collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have said Category is defined by an Item property, and only an Item comprising the Item property for said Category can be a member of said Category.

## Claim 35:

Regarding claim 35, Anglin and Nelson teaches a hardware/software interface system substantially as shown. Anglin and Nelson do not explicitly disclose an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. However, Chang teaches an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the

query are included in on the collection [referred to in the instant application as a category].)   It

would have been obvious to one with ordinary skill the art to combine the system as disclosed in

Anglin and Nelson with the disclosure of Chang because keeping the results of the query, all of

which exhibit a relationship, is useful in that the query will not have to be run again.  It is for this

reason that one of ordinary skill in the art would have been motivated to include an Item

comprising the Item property for one of said plurality of Categories is automatically a member of

that one of said plurality of Categories.


## Claim 36:

Regarding claim 36, Anglin and Nelson teaches a hardware/software interface system

substantially as shown.  Anglin and Nelson do not explicitly disclose an Item comprising one or

more Item properties corresponding to one or more Categories is automatically a member of each

such Categories having at least one of said corresponding Item properties.   However, Chang

teaches an Item comprising one or more Item properties corresponding to one or more Categories

is automatically a member of each such Categories having at least one of said corresponding

Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a

FederatedCollection allows an application program to process data objects resulting from a query

as a group or collection and at the same time preserves the sub-groupings relationships that exist

between them." By the nature of queries only returning the results that are related, the only

members of the Category will be from results that comprise the Item property.  Here, all of the

results of the query are included in on the collection [referred to in the instant application as a

category].)   It would have been obvious to one with ordinary skill the art to combine the system

as disclosed in Anglin with the disclosure of Chang because keeping the results of the query, all

of which exhibit a relationship, is useful in that the query will not have to be run again. It is for

this reason that one of ordinary skill in the art would have been motivated to include an Item

comprising one or more Item properties corresponding to one or more Categories is

automatically a member of each such Categories having at least one of said corresponding Item

properties.

**10.    Claims 23, 27, 28, and 30 rejected under 35 U.S.C. 103(a) as being unpatentable over**

**US patent application 2004/0199521 by Anglin et. al. (hereafter Anglin) and US Patent**

**7158962 by Nelson (hereafter Nelson) further in view of US Patent 6438545 by Beauregard**

**et. al. (hereafter Beauregard)**

## Claim 23:

Anglin and Nelson do not explicitly disclose "wherein said Item is a fundamental unit of

information manipulated by a virtual machine manager". However, Beauregard teaches said

Item is a fundamental unit of information manipulated by a virtual machine manager. (See

column 13, lines 12-16 "This broad I/O capability can be provided under the Virtual Machine

Manager (VMM) that is available under Win32. The VMM is an extensible operating system

whose core and standard components are provided by Microsoft Corporation.") Because of the

advantages provided by VMM as taught in Beauregard, such as the broad I/O capability, it would

have been obvious to one with ordinary skill in the art to combine the VMM of Beauregard with

the teaching of Anglin and Nelson. It is for this reason that one of ordinary skill in the art would

have been motivated to have said Item is a fundamental unit of information manipulated by a

virtual machine manager.


**Claim 27:**

Anglin discloses "The hardware/software interface system of claim 23 wherein said Item is not

owned by said Item Folder, such that the deletion of said Item Folder does not automatically

result in the deletion of said Item." (See page 3, paragraph [0028] "The secondary deletion

ensures that a storage object and corresponding storage object entry are only removed if the

storage object is not a member of any further groups after eliminating the relationship between

the storage object and group i." This follows the request to delete the "group leader" which

represents the group ID of the "storage group" or the "Item Folder" as in referred to in the

claim.)


**Claim 28:**

Anglin discloses "wherein said Item is automatically deleted when it no longer belongs to any

Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in

the associated groups fields for the specified storage object entry, then the storage management

server deletes the specified storage object entry from the storage database and deletes the

identifier of the deleted storage object entry from the group entry for the target group.")

**Claim 30:**

Anglin discloses "wherein said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

**11.    Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over US patent application 2004/0199521 by Anglin et. al. (hereafter Anglin) and US Patent 7158962 by Nelson (hereafter Nelson) further in view of "Typing a Multi-Language Intermediate Code" POPL 01' London, UK by Gordon et. al. (hereafter Gordon).**

**Claim 24:**

Anglin and Nelson do not explicitly disclose "The hardware/software interface system of claim 21 wherein said Item is a fundamental unit of information manipulated by a Common Language Runtime. " However, Gordon teaches said Item is a fundamental unit of information manipulated by a Common Language Runtime. (See conclusion p. 257 "One of the innovations in Microsoft's Common Language Runtime is support for typed stack pointers, for passing arguments and results by reference, for example. We presented formal typing rules and a type

safety result for a substantial fragment of Common Language Runtime intermediate language.

Our treatment of value types and pointer types appears to be new.") It would have been obvious

to one with ordinary skill in the art to combine the teaching of Anglin and Nelson with the

disclosure of Gordon because of the motivation of being able to pass the arguments and results

by reference to have more efficient processing of the data, less transfer overhead, and to be able

to handle more robust types of objects. It is for this reason that one of ordinary skill in the art

would have been motivated to have said Item is a fundamental unit of information manipulated

by a Common Language Runtime.

**12.     Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over US patent**

**application 2004/0199521 by Anglin et. al. (hereafter Anglin) and US Patent 7158962 by**

**Nelson (hereafter Nelson) further in view of US Patent 6430564 by Judge et. al. (hereafter**

**Judge)**

**<u>Claim 25:</u>**

Anglin and Nelson do not explicitly disclose "The hardware/software interface system of claim

21 wherein said Item is a fundamental unit of information manipulated by a Java Virtual

Machine". However, Judge teaches said Item is a fundamental unit of information manipulated

by a Java Virtual Machine. (See abstract "A data manager manages global data within a Java

Virtual Machine (JVM) installed and running in an embedded device. The data manager

maintains a data class list that stores data class identifiers associated with each data class object

currently loaded and cached in a data cache in the embedded device.") It would have been

obvious to one with ordinary skill in the art to combine Anglin and Nelson with Judge by using

the JVM to allow for more diverse types of objects to be processed by the hardware/software

interface system. It is for this reason that one of ordinary skill in the art would have been

motivated to have said Item is a fundamental unit of information manipulated by a Java Virtual

Machine.

13.    **Claims 29 and 31 is rejected under 35 U.S.C. 103(a) as being unpatentable over US**

**patent application 2004/0199521 by Anglin et. al. (hereafter Anglin) and US Patent 7158962**

**by Nelson (hereafter Nelson) and US Patent 6438545 by Beauregard et. al. (hereafter**

**Beauregard) further in view US 20040073560 by Edwards (hereafter Edwards)**

## Claim 29:

Anglin, Nelson, and Beauregard do not explicitly disclose "when it no longer belongs to any

Item Folder, automatically becomes a member of a default Item Folder." However, Edwards

teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a

member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to

'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or

delete a file, the file is passed to the operating system to be 'recycled'. This means it is not

deleted immediately, but stored in an area where it can be retrieved if required." In the instant

application, the area in which the item is stored is called the default Item Folder) It would have

been obvious to one with ordinary skill in the art to combine the Item and Item folder system of

Anglin, Nelson, and Beauregard with the disclosure of a recycling method of Edwards by simply

adding the recycling method of Edwards to the system of Anglin, Nelson, and Beauregard.

Edwards points out the advantage of keeping the potentially deleted file (or in the instant

application, the object,) available for future use rather than immediate deletion. It is for this

reason that one of ordinary skill in the art would have been motivated to have said Item, when it

no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.


## Claim 31:

Anglin, Nelson, and Beauregard do not explicitly disclose "said Item, when it is a member of

only one Item Folder and said Item Folder is deleted, automatically becomes a member of a

default Item Folder." However, Edwards teaches said Item, when it is a member of only one

Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item

Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than

delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is

passed to the operating system to be 'recycled'. This means it is not deleted immediately, but

stored in an area where it can be retrieved if required." In the instant application, the area in

which the item is stored is called the default Item Folder.) It would have been obvious to one

with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and

Beauregard with the disclosure of a recycling method of Edwards by simply adding the recycling

method of Edwards to the system of Anglin, Nelson, and Beauregard. Edwards points out the

advantage of keeping the potentially deleted file (or in the instant application, the object,)

available for future use rather than immediate deletion. It is for this reason that one of ordinary

skill in the art would have been motivated to have said Item, when it is a member of only one

Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item

Folder.

### Response to Arguments

14.·    Applicant's arguments with respect to claim 9/4/07 have been considered but are moot in

view of the new ground(s) of rejection.

Applicant's assert the following directed towards the Nelson reference:

A.  That the cited references do not disclose "each of said Items operable to belong to a plurality

of Item folders, wherein said item folders are themselves Items and said Items are configured to

be manipulated from any Item Folder they belong to" in claim 1 and similarly 21, 37, 43, 49, and

55.

*In response, the examiner respectfully disagrees.  Nelson discloses col. 1 lines 28-31, a*

*folder is a named collection of related items that can be retrieved, moved, and otherwise*

*manipulated as one entity.  Col. 3 lines 59-60, an item can be for example a folder or document.*

*Accordingly, each of said Items (col. 1 lines 28-31, items) operable to belong to a plurality of*

*Items Folders (col. 1 lines 28-31, collection of related items), wherein said Item Folders are*

*themselves Items (col. 3 lines 59-60, an folders are also Items) and said items are configured to*

be manipulated from any Item Folder they belong to (col. 1 lines 28-31, collection of related

items that can be retrieved, moved, and otherwise manipulated as one entity) is suggested.


In response to applicant's argument that the references fail to show certain features of

applicant's invention, it is noted that the features upon which applicant relies (i.e., that

manipulation is when an item accessed in one item folder, and revised, this revised item can then

be accessed directly from another item folder) are not recited in the rejected claim(s). Although

the claims are interpreted in light of the specification, limitations from the specification are not

read into the claims. See In re Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

## *Conclusion*

15.    The prior art made of record listed on PTO-892 and not relied, if any, upon is considered

pertinent to applicant's disclosure.


16.    Applicant's amendment necessitated the new ground(s) of rejection presented in this

Office action.  Accordingly, **THIS ACTION IS MADE FINAL**.  See MPEP § 706.07(a).

Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action.  In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action.  In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the date of this

final action.


## *Contact Information*

17.    Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Michael D. Pham whose telephone number is (571)272-3924.

The examiner can normally be reached on Monday - Friday  9am - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).


Michael Pham
Art Unit 2167
Examiner    M.P.


Luke S Wassum
Primary Examiner
Art Unit 2167